NHẬP MÔN HỌC MÁY

ĐỒ ÁN GIỮA KÌ

Người hướng dẫn:

TS. Trần Lương Quốc Đại

Người thực hiện:

Chung Thái Kiệt	52200140
Lê Hân	52200155
Huỳnh Thanh Bảo Ngọc	52200153

Nội dung

1. Giới thiệu dataset và bài toán
2. Xử lý dữ liệu
3. Đánh giá mô hình
4. Xử lý overfitting
5. Áp dụng feature selection

Giới thiệu tập dữ liệu

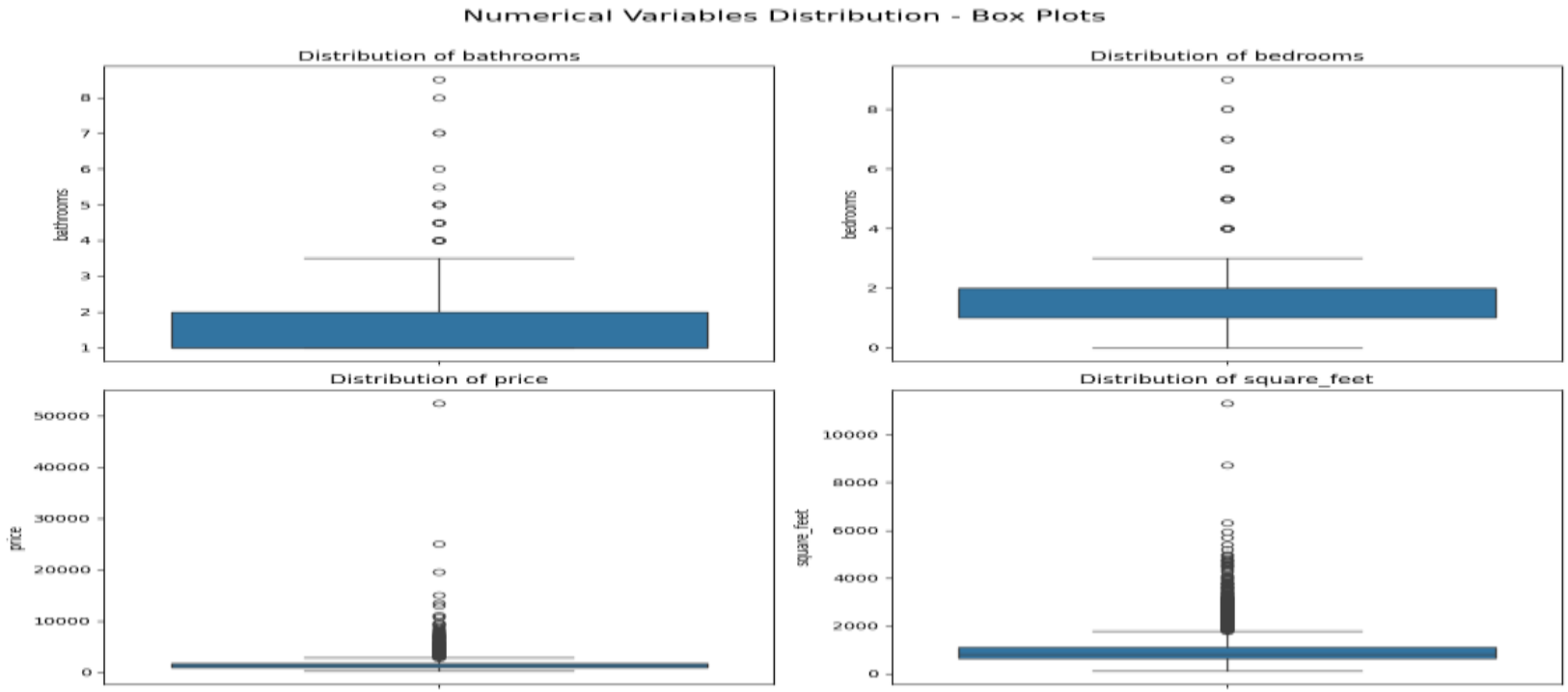
Dataset "Apartment for Rent Classified" từ UCI Machine Learning Repository cung cấp thông tin về các căn hộ cho thuê, được trích từ các trang web, nền tảng bất động sản của Mỹ. Bộ dữ liệu chứa 10000 hàng và 13 cột, dữ liệu gồm các thuộc tính sau:

- Amenities: categorical
- Bathrooms: float
- Bedrooms: float
- Has_photo: categorical
- Pet_allowed: categorical
- Price: integer
- Square_feet: : integer
- Address: categorical
- Cityname: categorical
- Latitude: float
- Longitude: float
- Source: categorical
- Time: integer

Giới thiệu bài toán

	Hồi quy	Phân loại
Mục tiêu	Dự đoán diện tích căn hộ dựa trên các đặc trưng như giá, số phòng tắm, phòng ngủ, tiện ích và vị trí địa lí,...	Dự đoán tiện ích (amenities) của một căn hộ dựa trên đặc trưng như giá, số phòng tắm, phòng ngủ, diện tích vị trí địa lí
Mô hình	Linear Regression, Random Forest, Decision Tree, Gradient Boosting	Logistic Regression, Decision Tree, K-Nearest Neighbors (KNN).
Ứng dụng	Tìm kiếm và ra quyết định khi mua bất động sản. Người dùng cung cấp ngân sách,vị trí,số phòng ...Sau đó hệ thống sẽ trả về diện tích ước tính của căn hộ	Tìm kiếm và ra quyết định khi mua bất động sản. Người dùng cung cấp ngân sách, vị trí, số phòng..Sau đó hệ thống sẽ xuất ra căn hộ loại tiện ích phù hợp

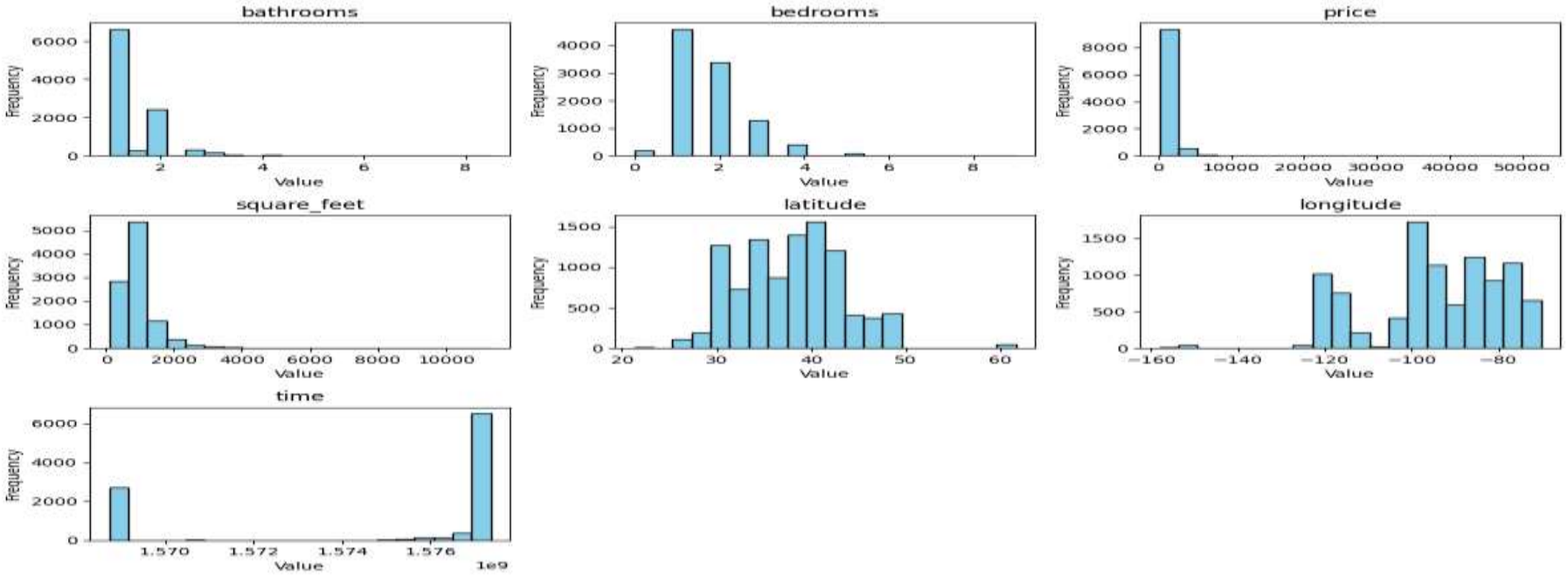
Data Visualization



Biểu đồ hộp với giá trị numerical

Cả bốn phân bố đều có ngoại lai, các giá trị trung vị (đường nằm giữa hộp của mỗi biểu đồ) cho thấy phần lớn bất động sản có số phòng tắm và phòng ngủ ít, giá thấp và diện tích nhỏ. Với một vài bất động sản lớn hoặc đắt hơn làm lệch phân bố.

Data Visualization

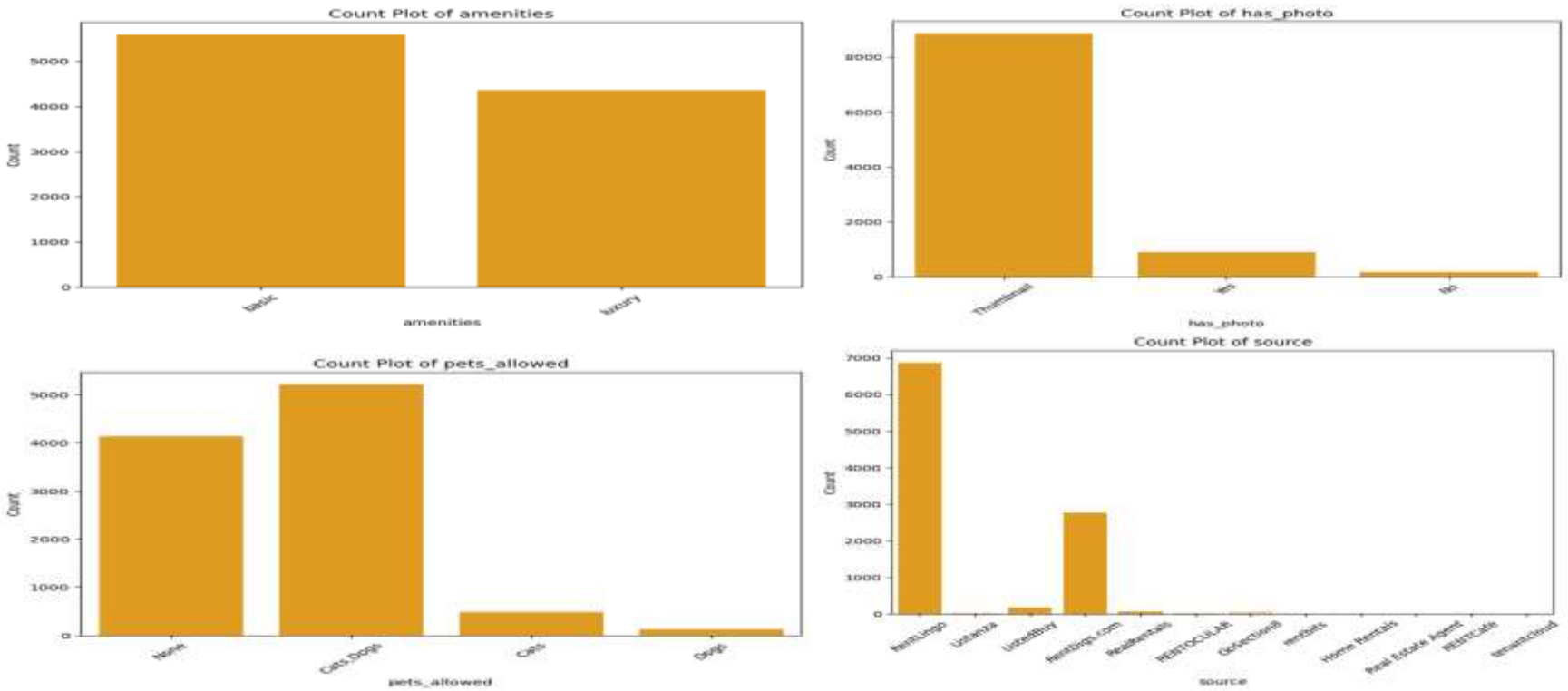


Biểu đồ tần suất với giá trị numerical

Các thuộc tính **Bathrooms**, **Bedrooms**, **Price**, và **Square Feet** đều lệch phải. Điều này ảnh hưởng đến việc tính trung bình và gây ra sai lệch khi đánh giá đặc điểm chung của các căn nhà.

Time lệch trái, cho thấy dữ liệu được thu thập chủ yếu trong một thời điểm nhất định, có thể ảnh hưởng đến các phân tích xu hướng theo thời gian.

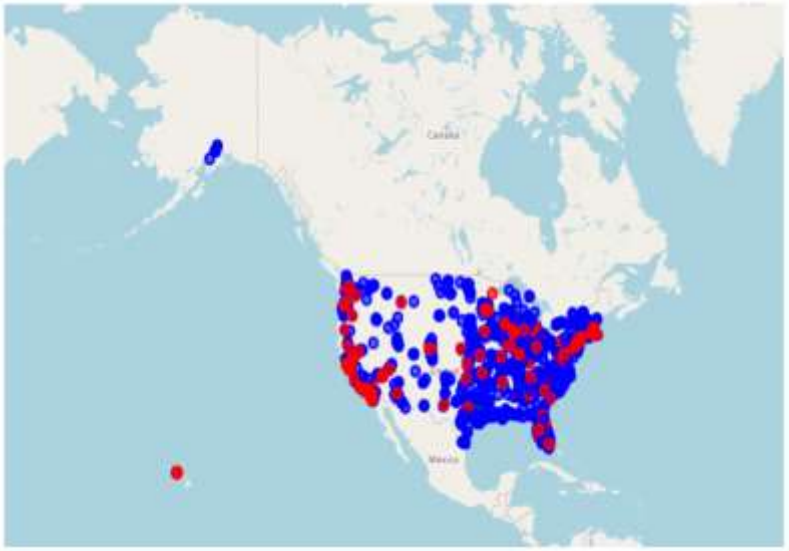
Data Visualization



Biểu đồ cột với giá trị categorical

Amenities tập trung ở basic, **has_photo** tập trung ở thumbnail, pet tập trung ở cats,dog, **source** tập trung ở RentLingo, RentDigs.com

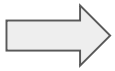
Data Visualization: Phân bố nhà dựa trên kinh độ, vĩ độ



Dữ liệu cho thấy phân bổ giá nhà cao tập trung vào các khu vực ven biển hoặc ở các thành phố lớn như **Chicago** và **Los Angeles**.

Tiền xử lý dữ liệu: Xử lý dữ liệu bị null

amenities	3549
bathrooms	34
bedrooms	7
pets_allowed	4163
cityname	77
state	77



amenities	0
bathrooms	0
bedrooms	0
pets_allowed	0
cityname	0
state	0

Amenities: chuyển dữ liệu null thành “basic”

Bathrooms, bedrooms: xóa các hàng có giá trị null ở các trường này

Pets_allowed: chuyển giá trị null thành “no”

Cityname, state: Với các giá trị null, gán giá trị của nó theo giá trị của kinh độ, vĩ độ

Tiền xử lý dữ liệu: Type conversion

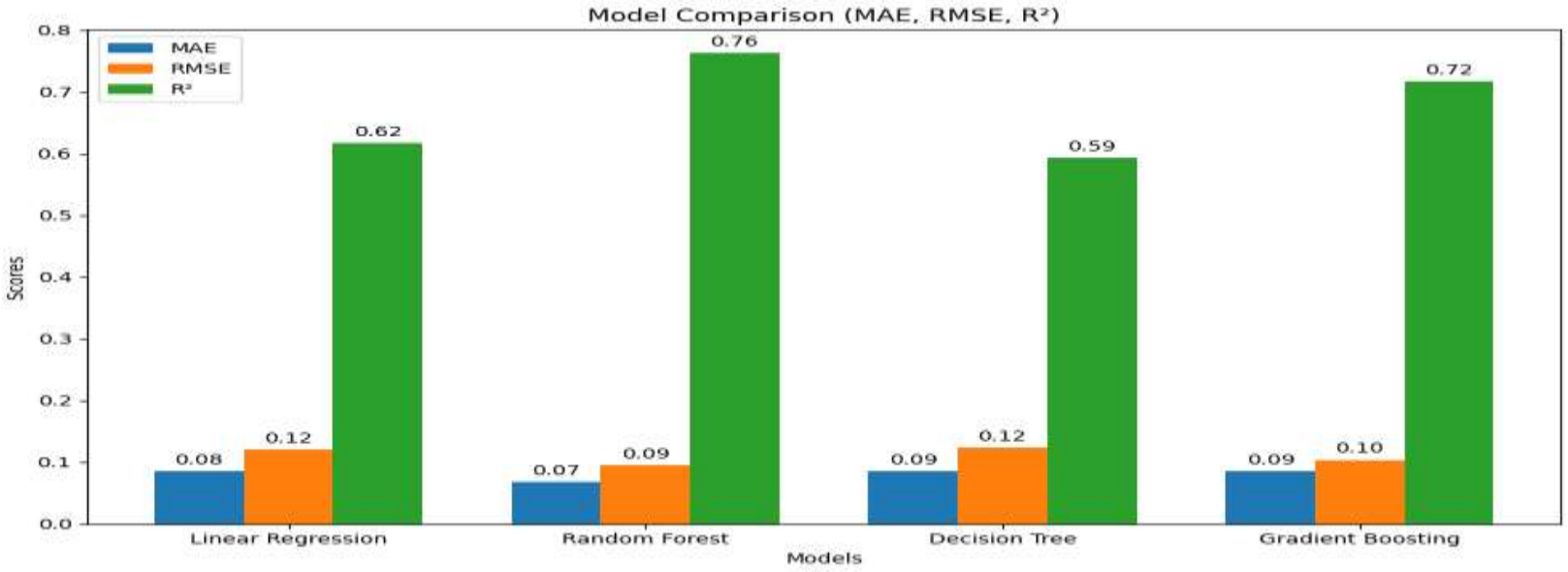
#	Column	Non-Null Count	Dtype
0	amenities	9949 non-null	object
1	bathrooms	9949 non-null	float64
2	bedrooms	9949 non-null	float64
3	has_photo	9949 non-null	object
4	pets_allowed	9949 non-null	object
5	price	9949 non-null	int64
6	square_feet	9949 non-null	int64
7	cityname	9949 non-null	object
8	state	9949 non-null	object
9	latitude	9949 non-null	float64
10	longitude	9949 non-null	float64
11	source	9949 non-null	object
12	time	9949 non-null	int64

#	Column	Non-Null Count	Dtype
0	bathrooms	9949 non-null	float64
1	bedrooms	9949 non-null	float64
2	has_photo	9949 non-null	int64
3	pets_allowed	9949 non-null	int64
4	price	9949 non-null	int64
5	square_feet	9949 non-null	int64
6	latitude	9949 non-null	float64
7	longitude	9949 non-null	float64
8	year	9949 non-null	int32
9	month	9949 non-null	int32
10	day_of_week	9949 non-null	int32
11	hour	9949 non-null	int32
12	is_weekend	9949 non-null	int32
13	amenities_basic	9949 non-null	bool
14	amenities_luxury	9949 non-null	bool
15	source_GoSection8	9949 non-null	bool
16	source_Home Rentals	9949 non-null	bool
17	source_Listanza	9949 non-null	bool
18	source_ListedBuy	9949 non-null	bool
19	source_RENTCafé	9949 non-null	bool
...			
25	source_rentbits	9949 non-null	bool
26	source_tenantcloud	9949 non-null	bool

Xây dựng model

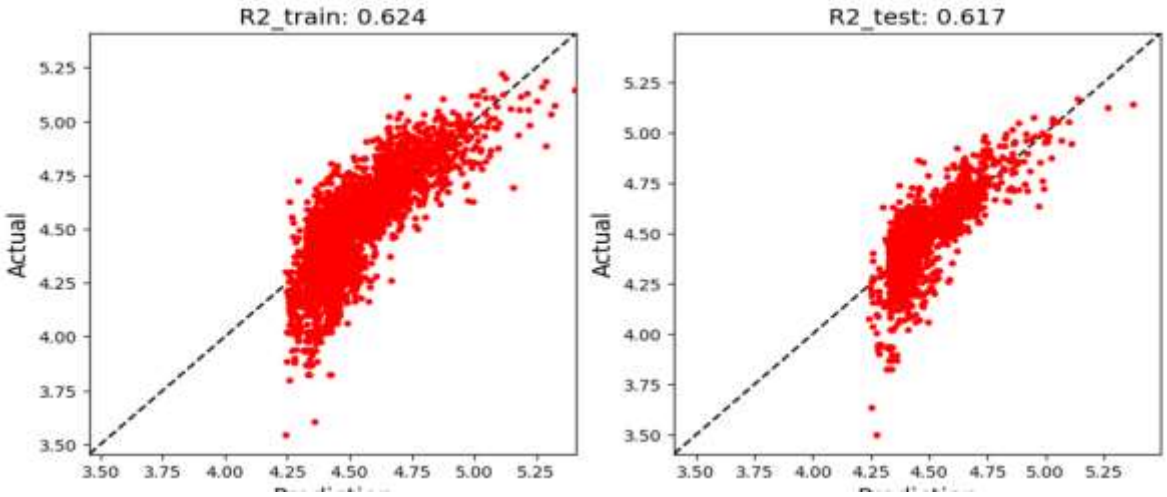
	Hồi quy	Phân loại
Mô hình	Linear Regression, Random Forest, Decision Tree, Gradient Boosting	Logistic Regression, Decision Tree, KNN
Tham số đánh giá	MAE, RMSE, R^2	Accuracy

Đánh giá kết quả và so sánh: Hồi quy



Nhận xét: Random Forest có hiệu suất tốt nhất với MAE và RMSE thấp nhất và R2 cao nhất, cho thấy nó phù hợp với dữ liệu

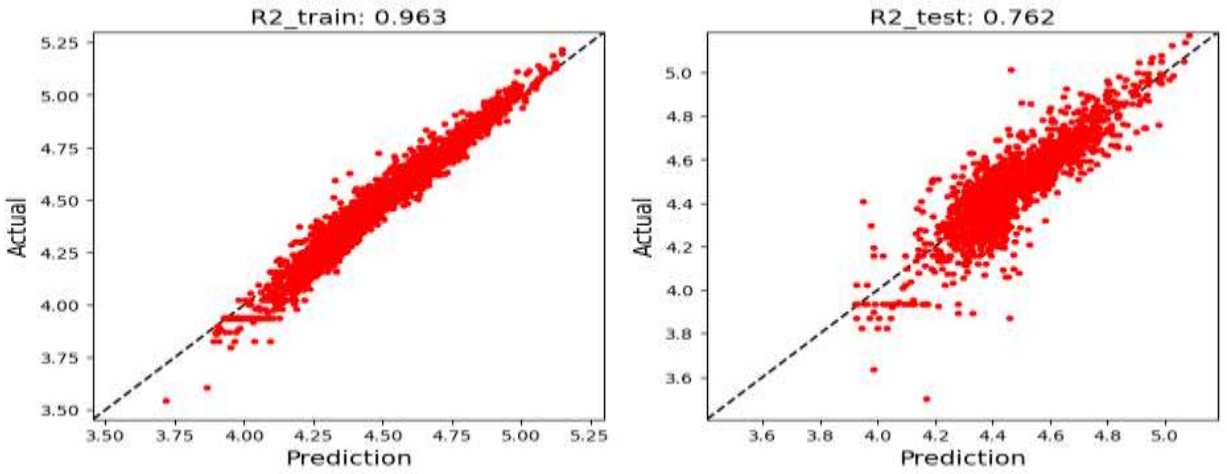
Sự phân tán của điểm của các điểm dự đoán: Linear Regression



Training R^2 = 0.624 và **Test R^2 = 0.617**: Giá trị R^2 trên tập huấn luyện và kiểm tra khá gần nhau, cho thấy mô hình tổng quát hóa tốt mà không bị overfitting.

Sự chênh lệch giữa hai giá trị là nhỏ, cho thấy mô hình có khả năng khái quát khá tốt, nhưng hiệu đều không cao, chỉ ở mức trung bình cho thấy mô hình chưa hoàn toàn nắm bắt được các yếu tố phức tạp trong dữ liệu

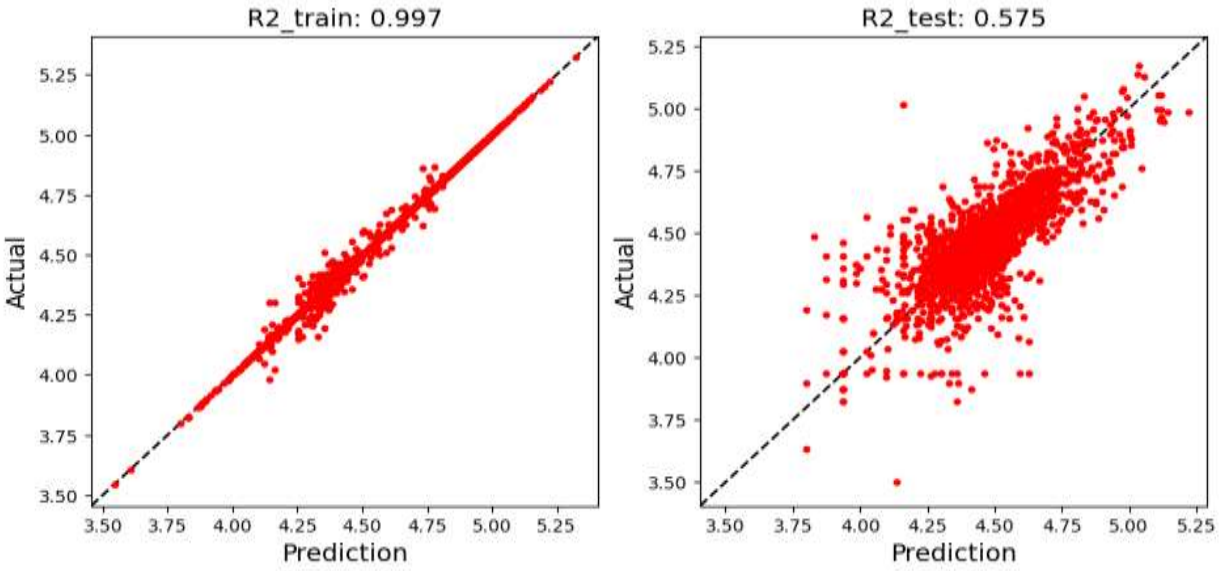
Sự phân tán của điểm của các điểm dự đoán: Random Forest



Training R^2 = 0.963 và **Test R^2 = 0.762**: Mô hình nắm bắt tốt các mẫu trong tập huấn luyện, nhưng sự chênh lệch lớn giữa R^2 trên tập huấn luyện và kiểm tra cho thấy mô hình có thể bị overfitting.

Mặc dù R^2 trên tập kiểm tra cao hơn so với hồi quy tuyến tính, nhưng sự khác biệt này chỉ ra rằng mô hình đang cố gắng quá mức để khớp với dữ liệu huấn luyện và có thể giảm khả năng tổng quát hóa trên dữ liệu mới.

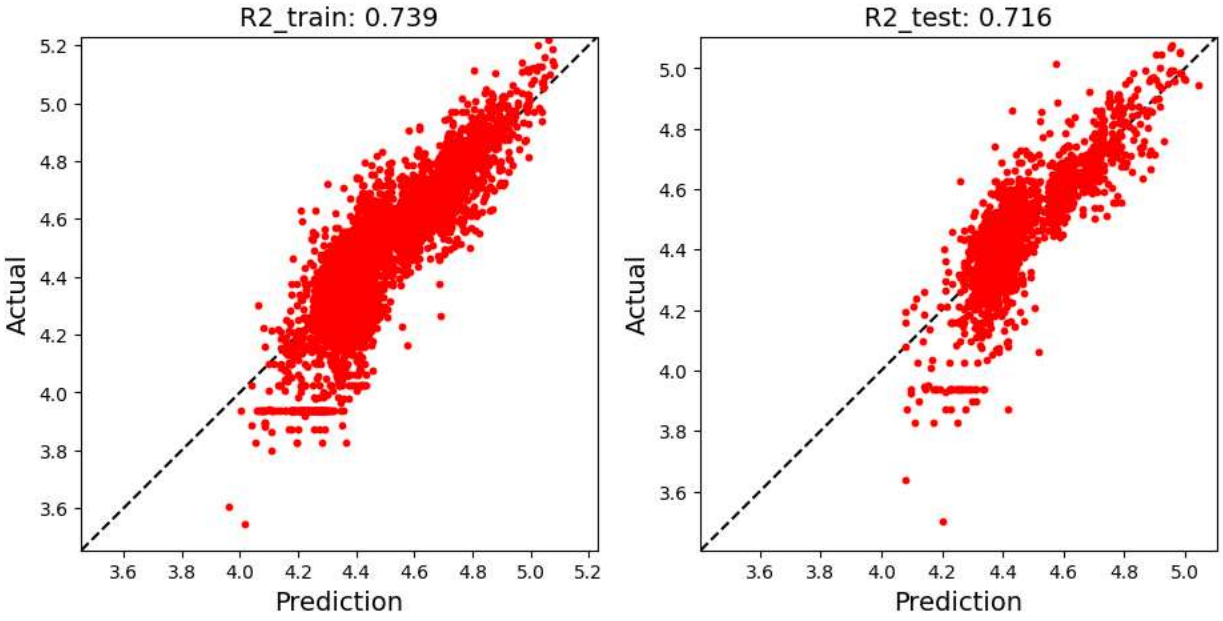
Sự phân tán của điểm của các điểm dự đoán: Decision Tree



Training R^2 = 0.997 và **Test R^2 = 0.575**: Mô hình có R^2 rất cao trên tập huấn luyện, cho thấy nó khớp cực kỳ tốt với dữ liệu huấn luyện.

Tuy nhiên, sự chênh lệch lớn giữa R^2 trên tập huấn luyện và tập kiểm tra (0.575) cho thấy mô hình bị **overfitting** nghiêm trọng. Mô hình học quá chi tiết từ dữ liệu huấn luyện, dẫn đến khả năng tổng quát hóa kém trên tập kiểm tra và dữ liệu mới.

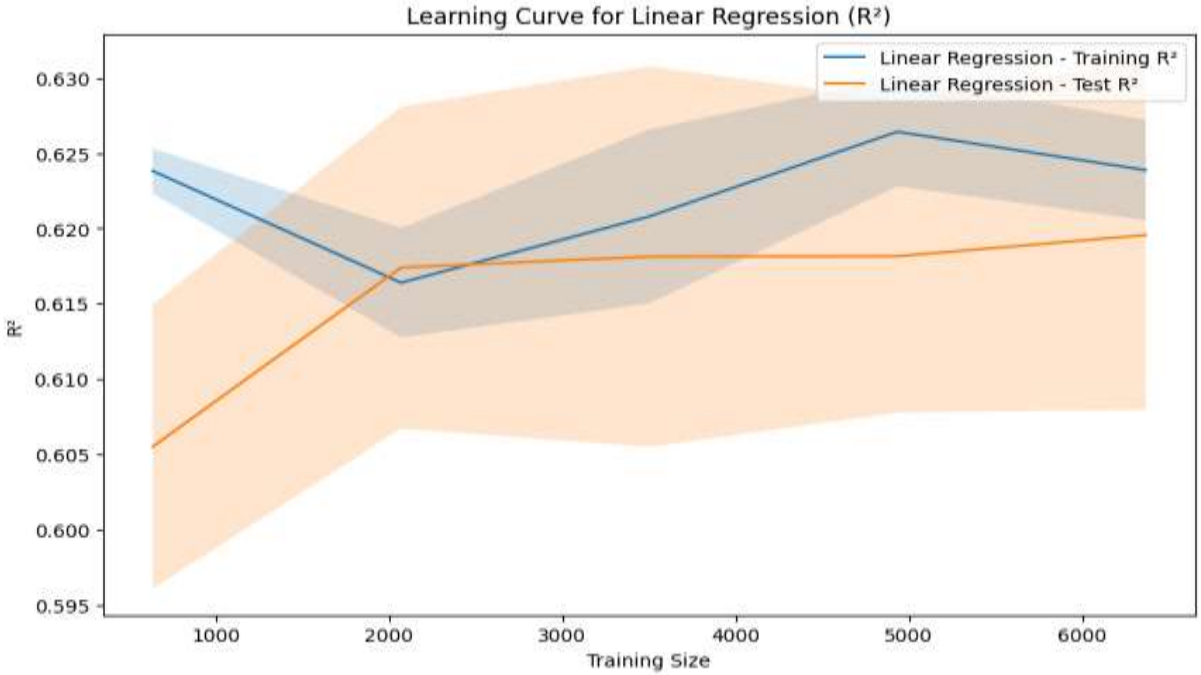
Sự phân tán của điểm của các điểm dự đoán: Gradient Boosting



Training R^2 = 0.739 và **Test R^2 = 0.716**: Mô hình có khả năng tổng quát hóa tốt với sự chênh lệch nhỏ giữa tập huấn luyện và kiểm tra.

Không bị overfit, và hiệu suất ổn định trên cả hai tập dữ liệu, vượt trội hơn so với các mô hình như Cây quyết định và Rừng ngẫu nhiên

Learning Curve for Linear Regression

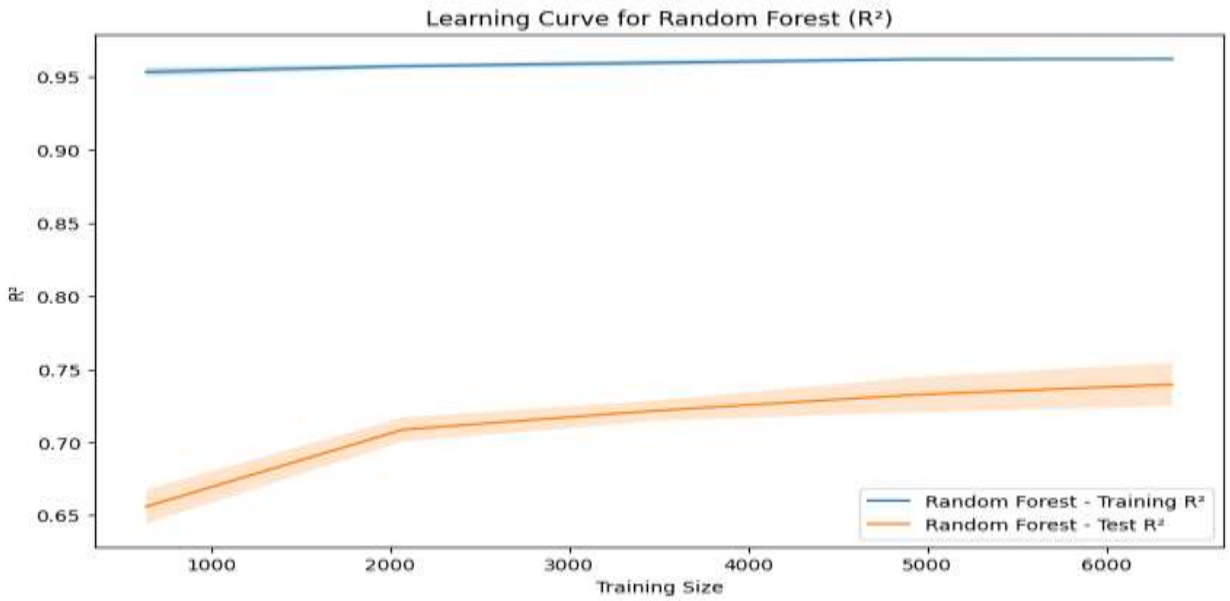


Training R^2: giảm nhẹ và ổn định ở 0.620, cho thấy khả năng khái quát hóa

Test R^2: tăng dần và ổn định quanh 0.615, chứng tỏ mô hình khái quát với dữ liệu lớn

Overfitting: Khoảng cách giữa Training R^2 và Test R^2 nhỏ, hai đường hội tụ và ổn định khi dữ liệu tăng, cho thấy mô hình không bị overfitting.

Learning Curve for Random Forest

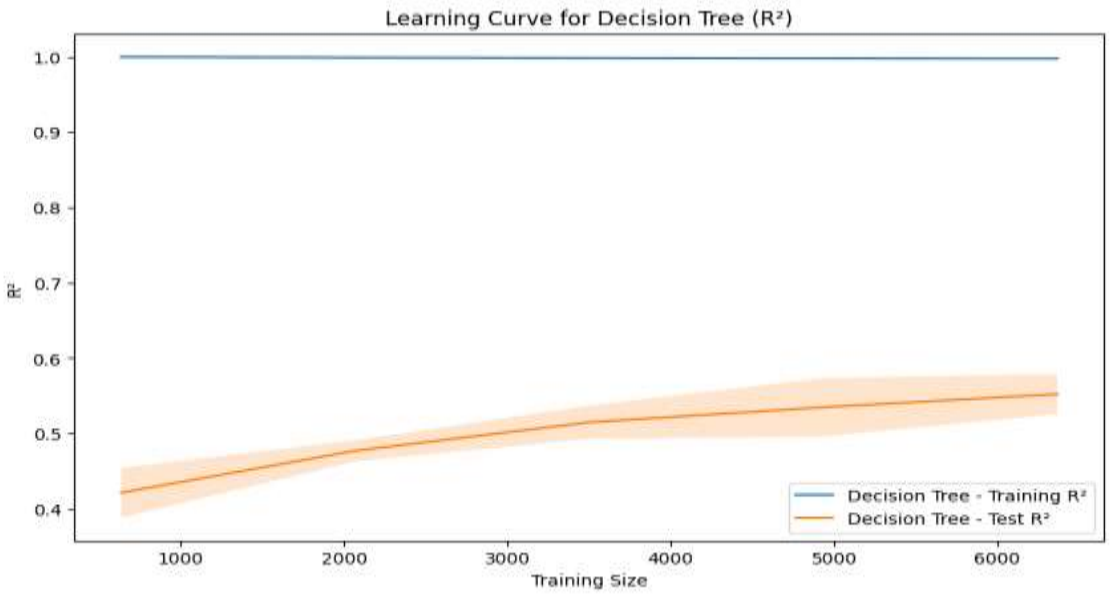


Training R^2: Rất cao (~0.95-0.96) và ổn định, cho thấy mô hình khớp tốt với dữ liệu huấn luyện.

Test R^2: Ban đầu thấp (~0.65), tăng dần và ổn định ở ~0.75 khi tập huấn luyện lớn hơn, nhưng vẫn thấp hơn nhiều so với Training R^2.

Overfitting: Khoảng cách lớn giữa Training R^2 và Test R^2 cho thấy mô hình bị overfitting, với khả năng khái quát kém trên tập kiểm tra.

Learning Curve for Decision Tree

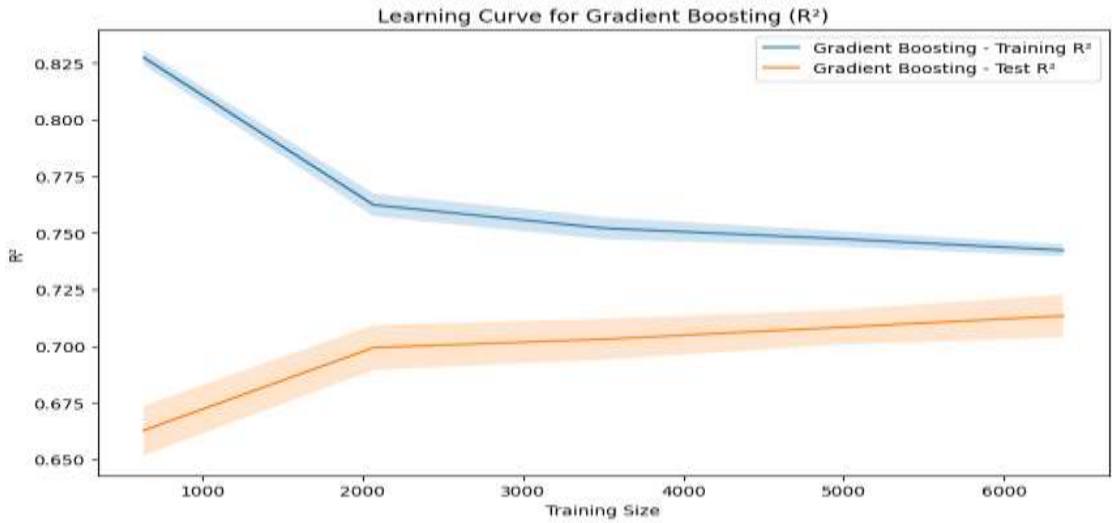


Training R^2: Duy trì ở mức 1, cho thấy mô hình hoàn toàn khớp với dữ liệu huấn luyện, dấu hiệu của overfitting.

Test R^2: Tăng chậm và chỉ đạt khoảng 0.5, chứng tỏ mô hình khái quát kém trên tập kiểm tra.

Overfitting: Mô hình có dấu hiệu overfitting rõ ràng, đạt $R^2 = 1$ trên tập huấn luyện nhưng chỉ $R^2 = 0.5$ trên tập kiểm tra, cho thấy mô hình không khái quát tốt.

Learning Curve for Gradient Boosting

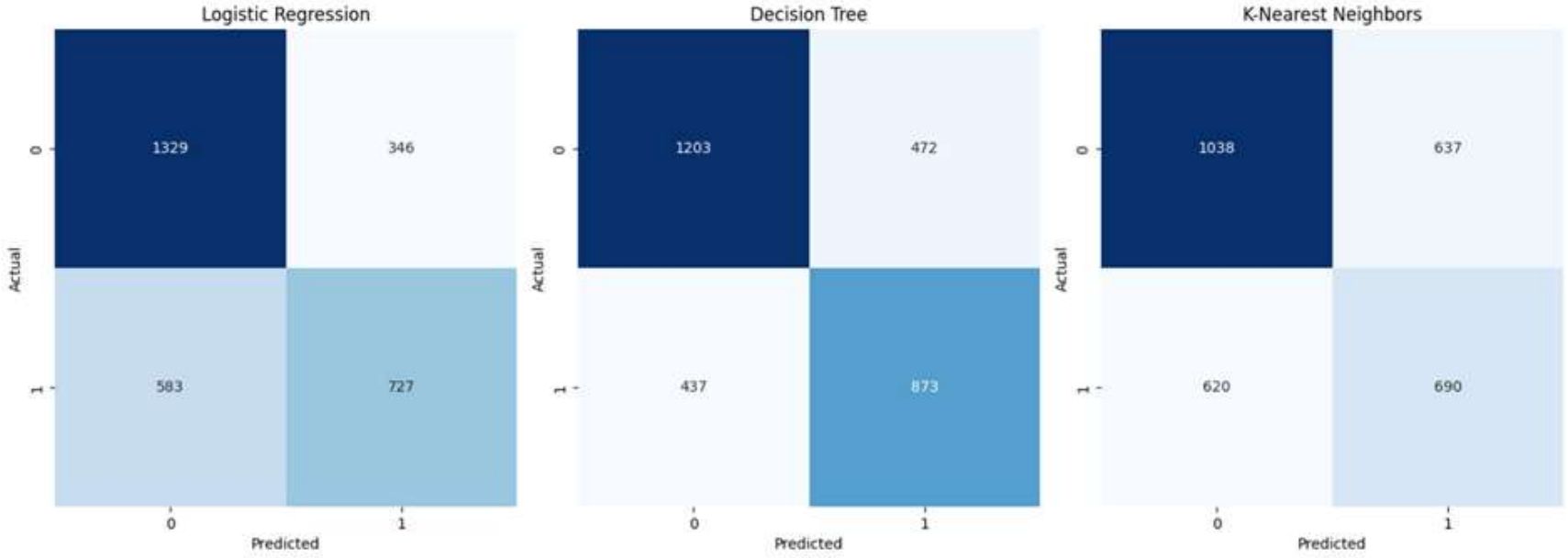


Training R²: Ban đầu cao (~0.825), nhưng giảm dần và ổn định ở mức ~0.75, cho thấy mô hình học tốt mà không bị overfitting quá mức.

Test R²: Tăng dần và ổn định quanh mức ~0.7, tiến gần giá trị của Training R², cho thấy khả năng tổng quát hóa tốt trên dữ liệu chưa thấy trước.

Overfitting: Không có dấu hiệu overfitting rõ ràng. Khoảng cách nhỏ và ổn định giữa Training R² và Test R² cho thấy mô hình tổng quát tốt, nhưng có thể cải thiện thêm qua tinh chỉnh siêu tham số hoặc tăng dữ liệu.

Đánh giá kết quả theo mô hình: Phân loại

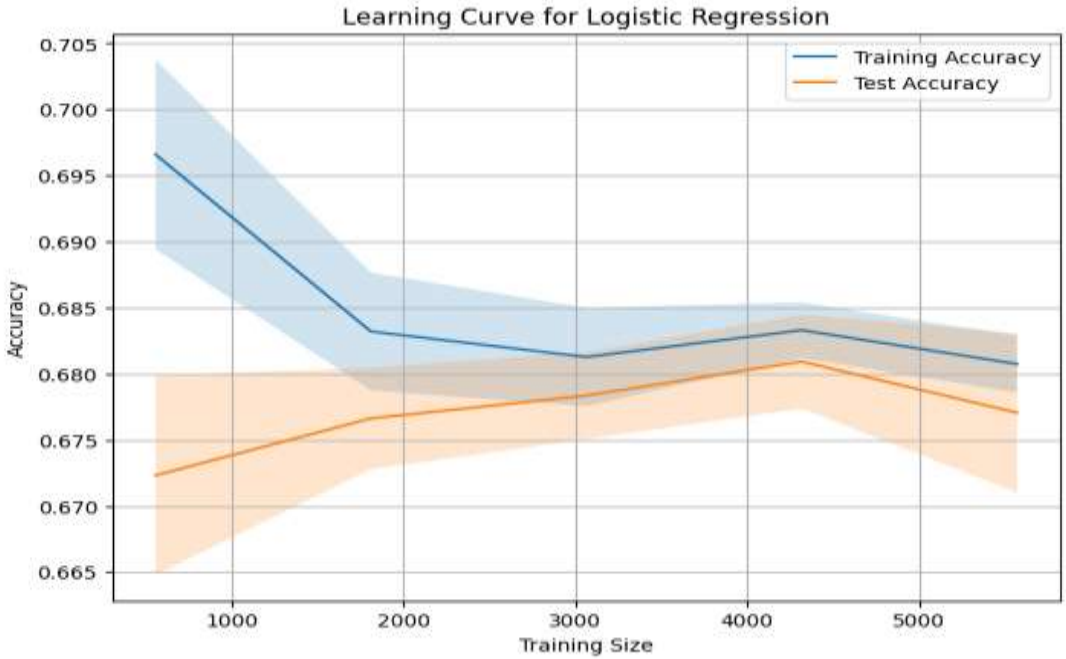


Logistic Regression: mô hình nghiêng về dự đoán 0

Decision Tree: có khả năng cân bằng hơn giữa hai lớp 0 và 1 so với Logistic Regression.

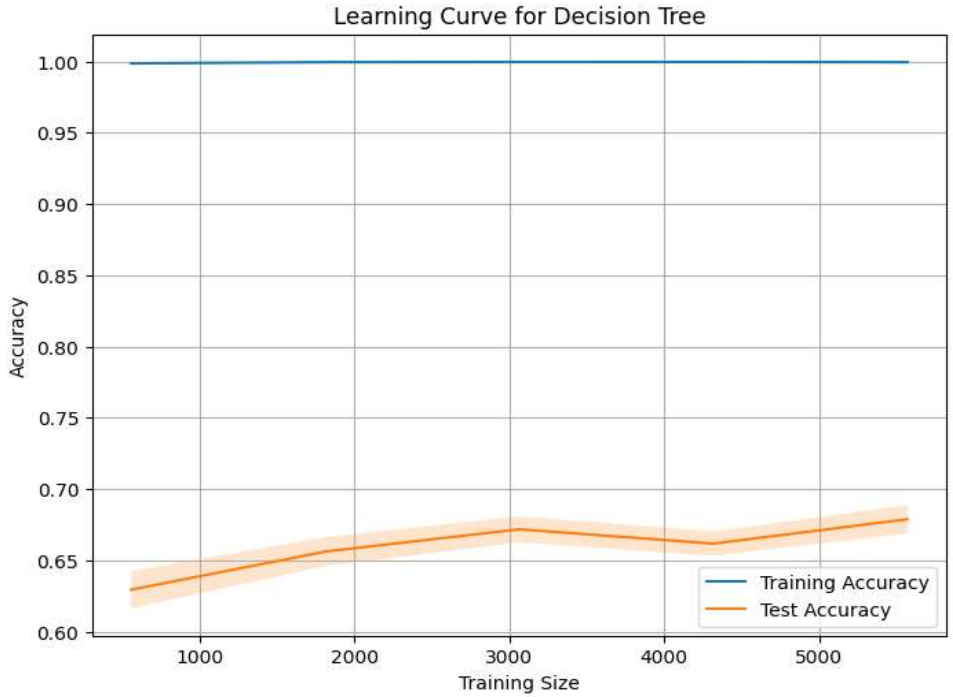
KNN: có tỷ lệ dự đoán đúng thấp hơn ở cả hai lớp so với Logistic Regression và Decision Tree.

Learning Curve for Logistic Regression



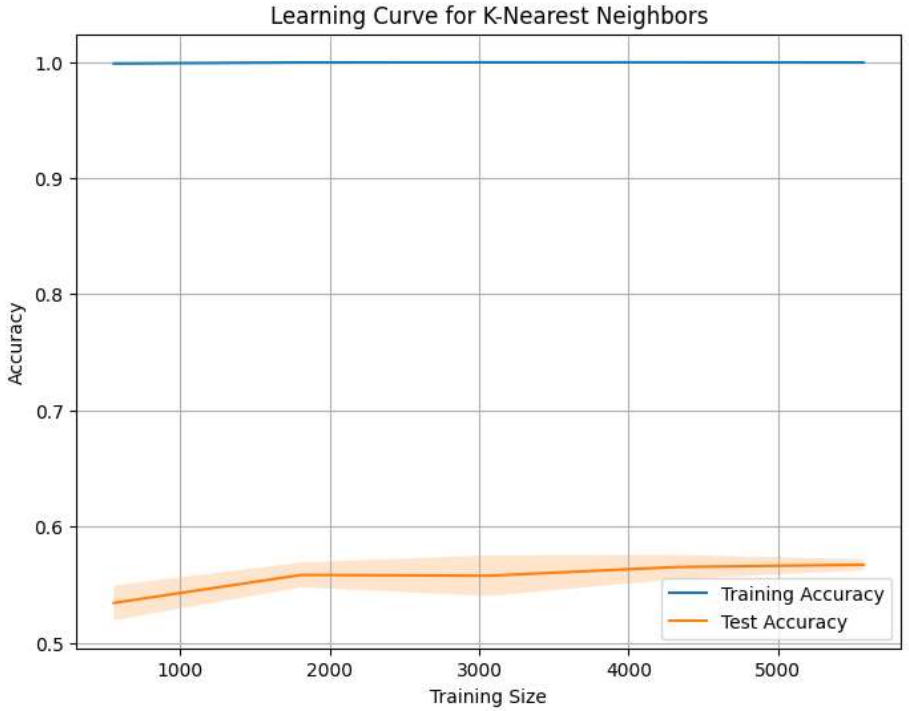
Hai đường accuracy của tập huấn luyện và tập kiểm tra khá gần nhau và ổn định khi kích thước tập huấn luyện tăng. Khoảng cách giữa chúng không quá lớn, cho thấy mô hình không bị overfitting hay underfitting, và có khả năng tổng quát tốt trên dữ liệu chưa thấy.

Learning Curve for Decision Tree



Khoảng cách giữa accuracy của tập huấn luyện và tập kiểm tra là dấu hiệu rõ ràng của overfitting. Mô hình học quá chi tiết trên dữ liệu huấn luyện nhưng không khái quát tốt trên dữ liệu mới. Để cải thiện, có thể giảm độ sâu của cây quyết định hoặc áp dụng kỹ thuật regularization để tăng khả năng tổng quát và giảm overfitting.

Learning Curve for KNN



Khoảng cách rõ rệt giữa accuracy của tập huấn luyện và tập kiểm tra là dấu hiệu của overfitting. Mô hình học rất tốt trên dữ liệu huấn luyện nhưng không khái quát tốt trên dữ liệu kiểm tra. Để cải thiện, có thể áp dụng các kỹ thuật regularization hoặc giảm độ phức tạp của mô hình để cải thiện khả năng tổng quát.

Xử lý Overfitting

Giới thiệu Overfitting

Định nghĩa Overfitting

- Overfitting là tình trạng mô hình học máy quá khớp với dữ liệu huấn luyện, đến mức không thể tổng quát hóa tốt trên dữ liệu mới.
- Mô hình trở nên quá phức tạp, chỉ có thể làm tốt trên tập dữ liệu huấn luyện nhưng kém hiệu quả trên tập dữ liệu kiểm tra/dự báo.

Lý do dẫn đến Overfitting

- **Mô hình quá phức tạp (over-parameterized)**
  - Mô hình có quá nhiều tham số so với lượng dữ liệu huấn luyện.
  - Mô hình có thể "nhớ" và phù hợp với các đặc điểm riêng biệt của dữ liệu huấn luyện quá tốt.
- **Dữ liệu huấn luyện không đủ**
  - Lượng dữ liệu huấn luyện quá ít so với độ phức tạp của mô hình.
  - Mô hình không thể học tổng quát từ lượng dữ liệu ít ỏi.

Giới thiệu Overfitting

Lý do dẫn đến Overfitting

- **Noise trong dữ liệu huấn luyện**
  - Dữ liệu huấn luyện chứa nhiều noise (nhiễu) hoặc outliers.
  - Mô hình cố gắng "học" cả nhiễu, dẫn đến overfitting.

Biểu hiện của overfitting

- Hiệu suất trên tập huấn luyện rất cao, nhưng hiệu suất trên tập kiểm tra/dữ liệu mới lại thấp.
- Mô hình có độ phức tạp quá cao so với lượng dữ liệu huấn luyện.
- Mô hình có khả năng "ghi nhớ" quá tốt các mẫu huấn luyện nhưng không thể tổng quát hóa.

Giải pháp xử lý Overfitting

- Giảm độ phức tạp của mô hình
  - Sử dụng các kỹ thuật regularization như L1, L2, Dropout,...
  - Cắt tỉa cây (tree pruning) trong các mô hình cây quyết định.
  - Giới hạn số lượng tham số của mô hình.
- Tăng lượng dữ liệu huấn luyện
  - Thu thập thêm dữ liệu mới.
  - Sử dụng data augmentation để tạo thêm dữ liệu tổng hợp.
- Sử dụng kỹ thuật cross-validation: Đánh giá mô hình một cách chính xác hơn trên các tập dữ liệu khác nhau.Giúp phát hiện overfitting sớm hơn.
- Điều chỉnh siêu tham số (Hyperparameter Tuning): Sử dụng các kỹ thuật như Grid Search hoặc Random Search để tìm các siêu tham số tối ưu cho mô hình

Kết quả mô hình sau khi chỉnh sửa Overfitting

Hồi quy

Sử dụng hyperparameter tuning, cross-validation, regularization vào model

Mô hình hồi quy : Random Forest Regressor

Sau khi áp dụng GridSearchCV để tìm hyper parameter, ta có được bộ tham số:

'bootstrap': True,

'max_depth': 10,

'min_samples_leaf': 2,

'min_samples_split': 2,

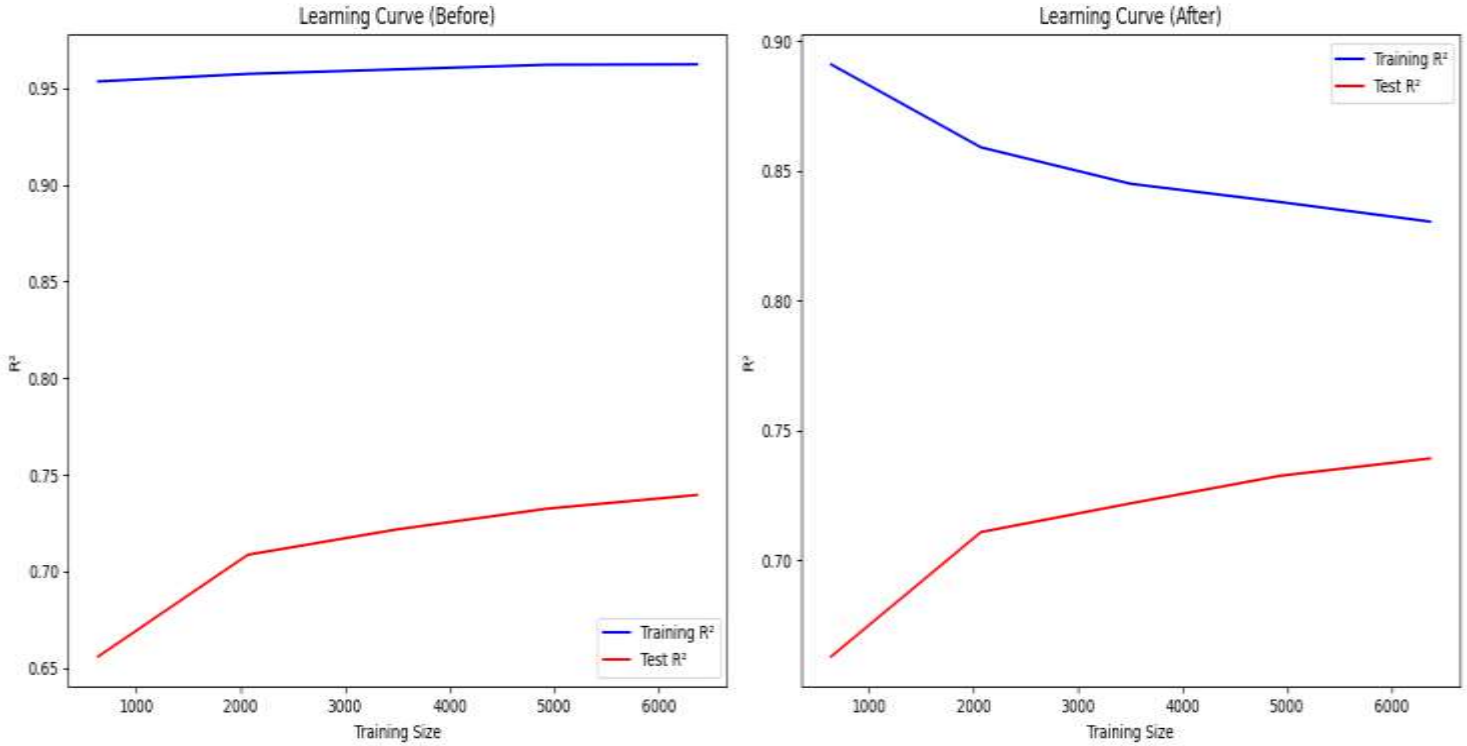
'n_estimators': 200,

'ccp_alpha': 0.0

	Metric	Trước overfitting	Sau overfitting
0	MAE	0.067292	0.069487
1	RMSE	0.094621	0.096317
2	R^2	0.761969	0.753357

Sự khác nhau giữa các tham số trước và sau khi xử lí không có sự thay đổi nhiều

Mô hình hồi quy: Random Forest Regressor



Trước xử lí có khoảng cách lớn giữa train và test, sau xử lí thì khoảng cách đã được cải thiện, cho thấy mô hình có xu hướng học tập tổng quát hơn dự đoán trên dữ liệu mới tốt hơn.

Mô hình hồi quy: Decision Tree Regressor

Sau khi áp dụng GridSearchCV để tìm hyper parameter, ta có được bộ tham số:

'max_depth':10,

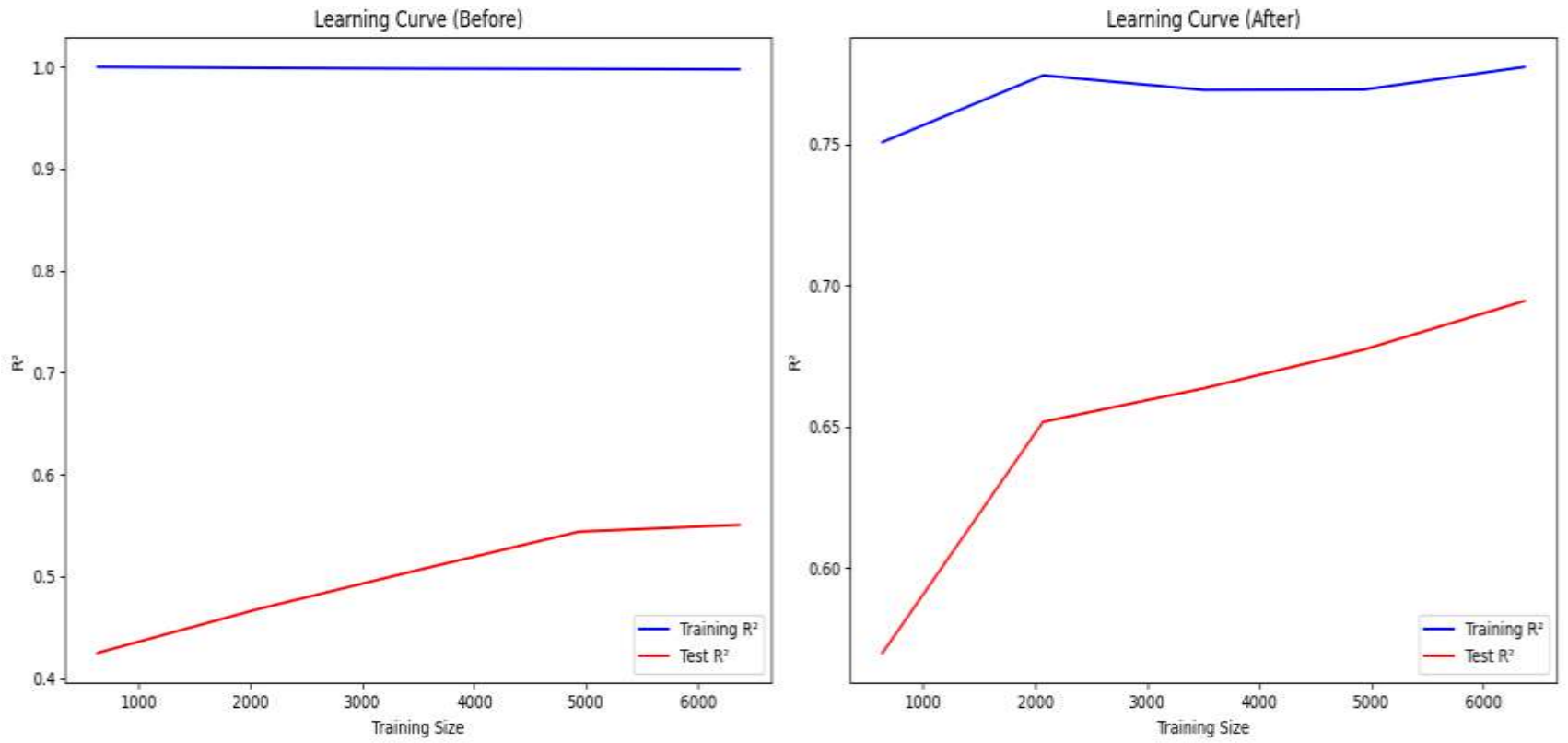
'min_samples_split':2,

'min_samples_leaf':10,

'max_features': None

	Metric	Trước overfitting	Sau overfitting
0	MAE	0.084822	0.072752
1	RMSE	0.123314	0.102990
2	R^2	0.595718	0.717997

Mô hình hồi quy: Decision Tree Regressor



Trước khi xử lý, mô hình quá khớp với dữ liệu huấn luyện, thể hiện qua R^2 chênh lệch lớn giữa tập huấn luyện và kiểm tra. Sau khi xử lí cho thấy mô hình đã tổng quát hóa tốt hơn và dự đoán dữ liệu mới chính xác hơn.

Mô hình hồi quy: Gradient Boosting Regressor

Sau khi áp dụng GridSearchCV để tìm hyper parameter, ta có được bộ tham số:

'n_estimators':200,

'learning_rate':0.2,

'max_depth':5,

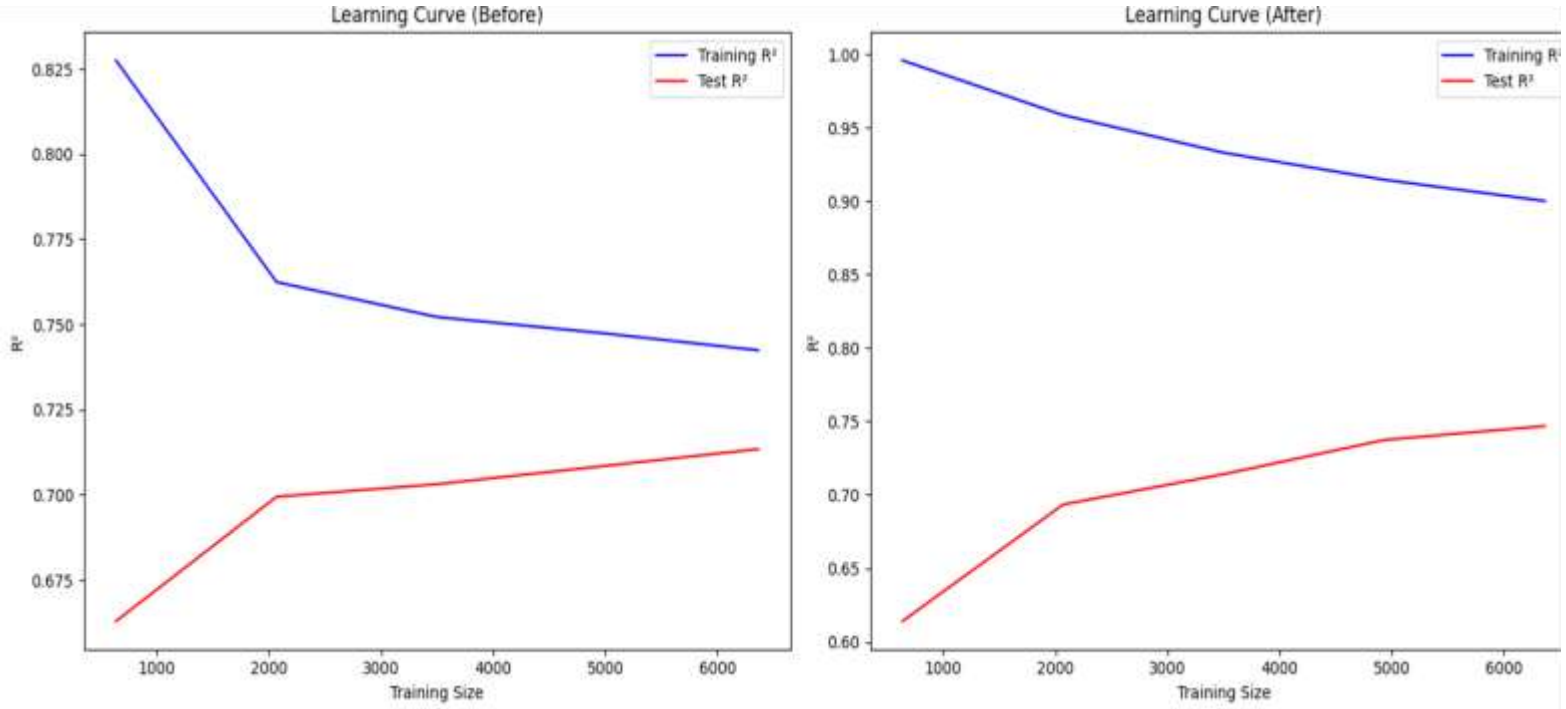
'min_samples_split':5,

'min_samples_leaf':2,

'subsample': 0.8

	Metric	Trước overfitting	Sau overfitting
0	MAE	0.073920	0.067693
1	RMSE	0.103331	0.093876
2	R^2	0.716130	0.765703

Mô hình hồi quy: Gradient Boosting Regressor



Trước khi xử lý, cho thấy mô hình học quá nhiều chi tiết không cần thiết.

Sau khi xử lý, cho thấy mô hình tổng quát tốt hơn và chính xác hơn trên dữ liệu mới.

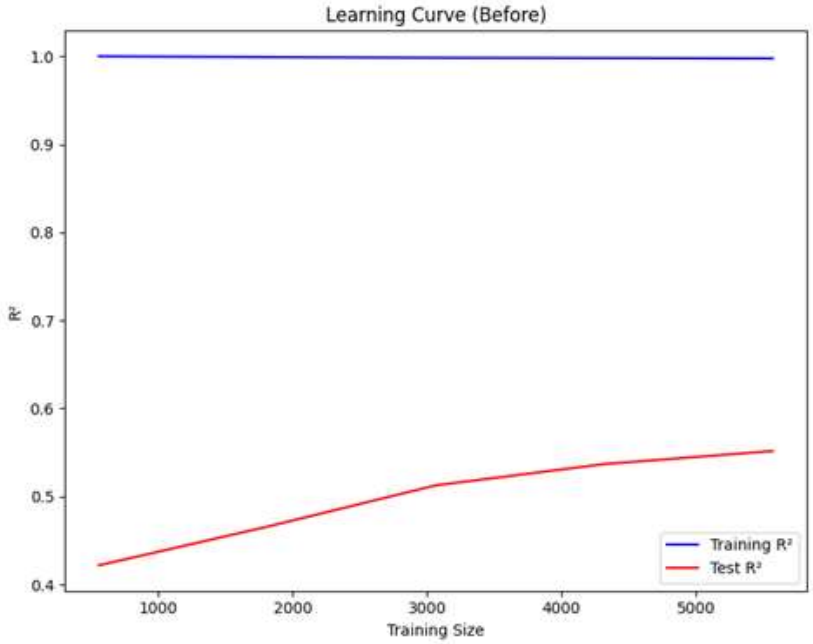
Phân loại

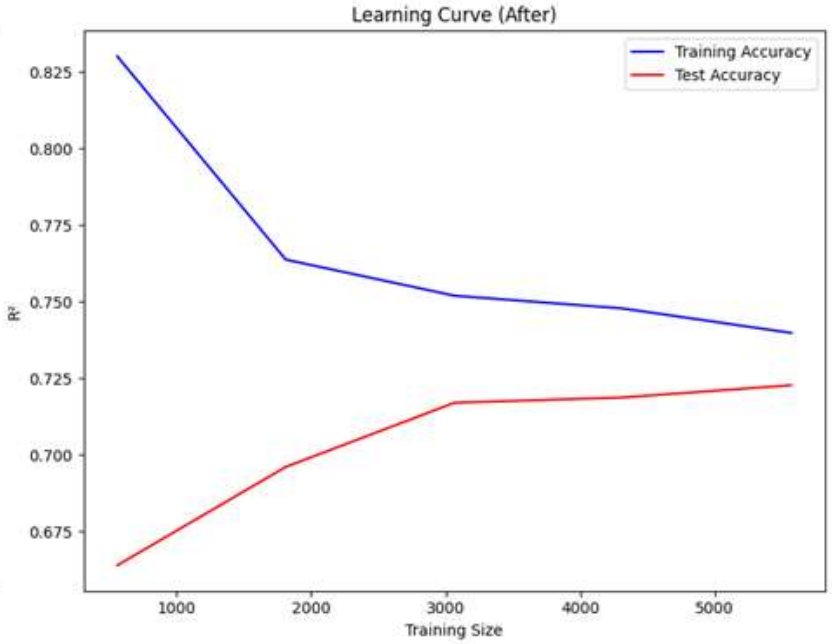
Sử dụng hyperparameter tuning, cross-validation, regularization vào model

Mô hình phân loại: Decision Tree Classifier

Sau khi áp dụng GridSearchCV để tìm hyper parameter, ta có được bộ tham số:

class_weight=None,criterion='gini',max_depth=None,
max_features=Nonemax_leaf_nodes=30,min_impurity_decrease=0.0,
min_samples_leaf=1 min_samples_split=2 splitter='best'





Mô hình phân loại: Decision Tree Classifier

Sử dụng model mới cho phân loại

Dùng thêm XGBClassifier với bộ tham số

n_estimators= 200,

max_depth= 6,

learning_rate= 0.1,

subsample= 0.8,

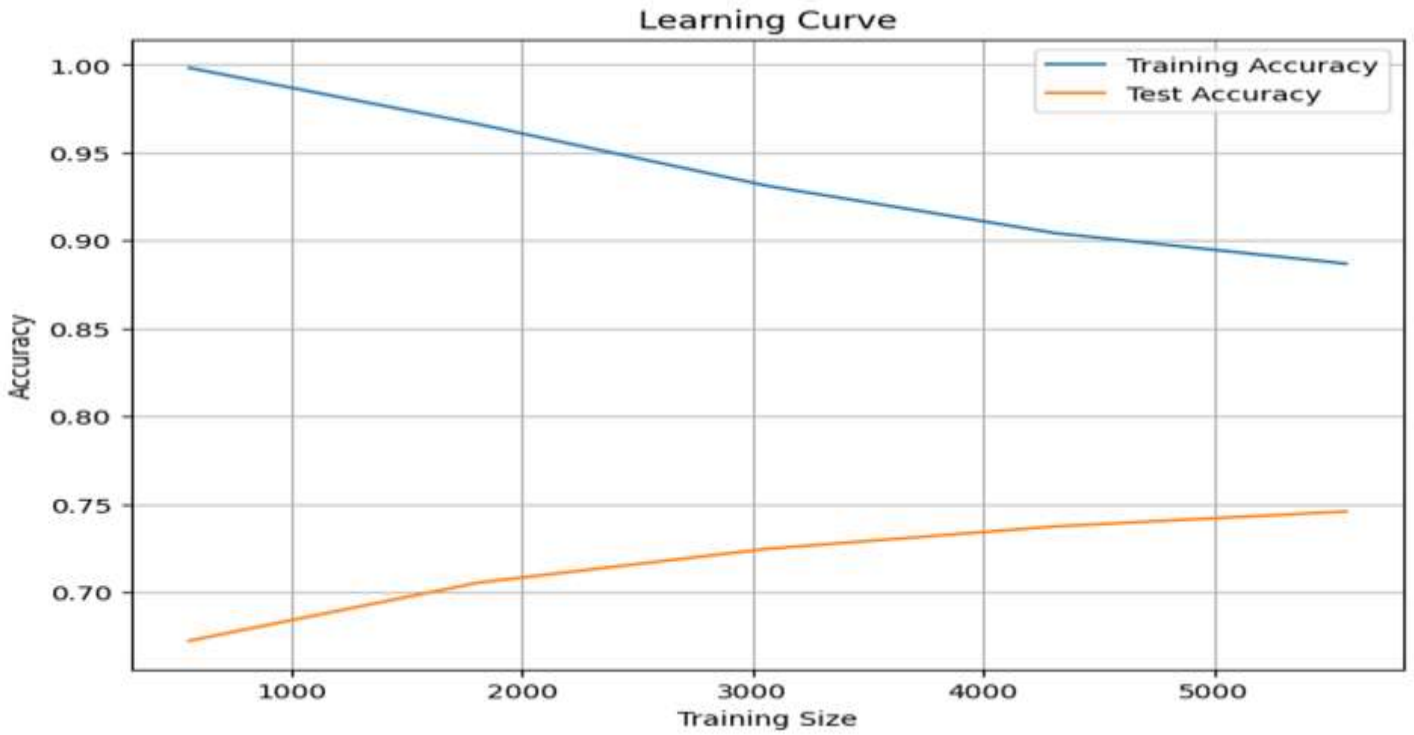
colsample_bytree= 0.8,

gamma= 0

Model	Accuracy
Logistic Regression	0.68
Decision Tree	0.69
KNN	0.57
XGBClassifier	0.76

=> XGBClassifier có accuracy cao hơn 3 model còn lại

Mô hình phân loại: XGBClassifier



Accuracy của train giảm, nó có xu hướng học tập và tổng quan dữ liệu. Do đó tại tập test có độ chính xác cao và dự đoán cho dữ liệu mới tốt hơn các model cũ

CHƯƠNG 3: FEATURE SELECTION USING CORRELATION ANALYSIS

GIỚI THIỆU VỀ FEATURE SELECTION

Trong học máy, lựa chọn đặc trưng là quá trình lựa chọn một tập hợp con các đặc trưng có liên quan (biến, dự đoán) để sử dụng trong xây dựng mô hình. Các kỹ thuật lựa chọn đặc trưng được sử dụng vì một số lý do:

- Tránh tính đa chiều
- Thời gian đào tạo ngắn
- Cải thiện khả năng tương thích của dữ liệu
- Đơn giản hóa các mô hình

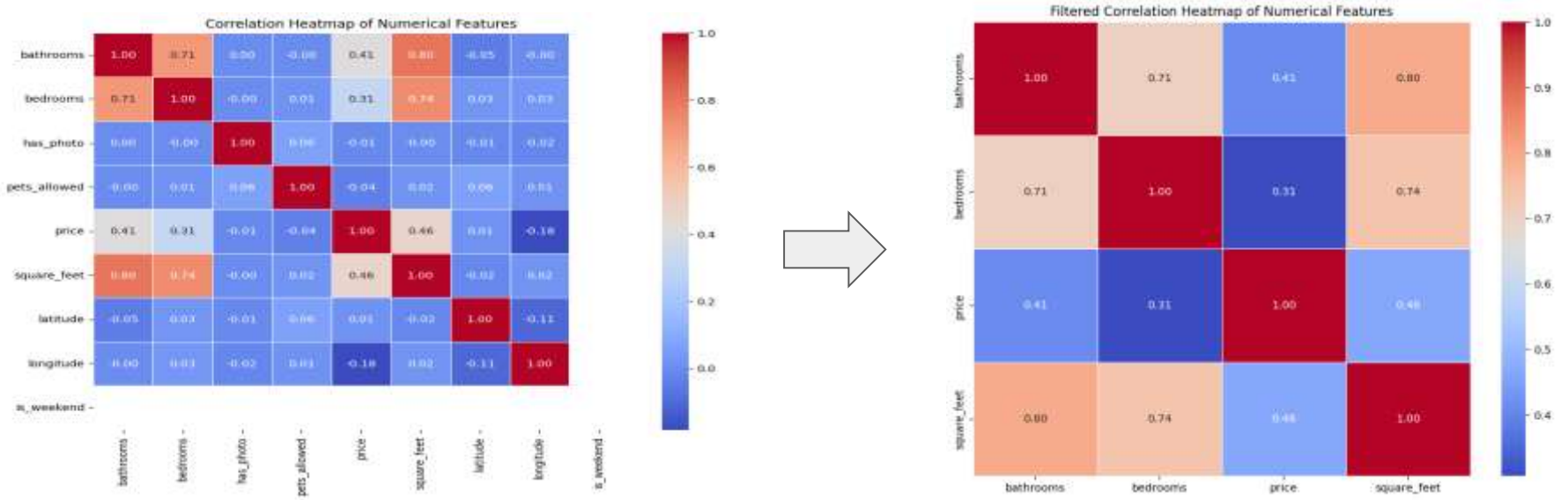
GIỚI THIỆU VỀ CORRELATION ANALYSIS

Correlation Analysis là một phương pháp trong lựa chọn tính năng (feature selection), thuộc nhóm **Filter Methods**, giúp xác định mối quan hệ giữa các đặc trưng trong bộ dữ liệu.

Khi phân tích tương quan có 2 trường hợp cần chú ý

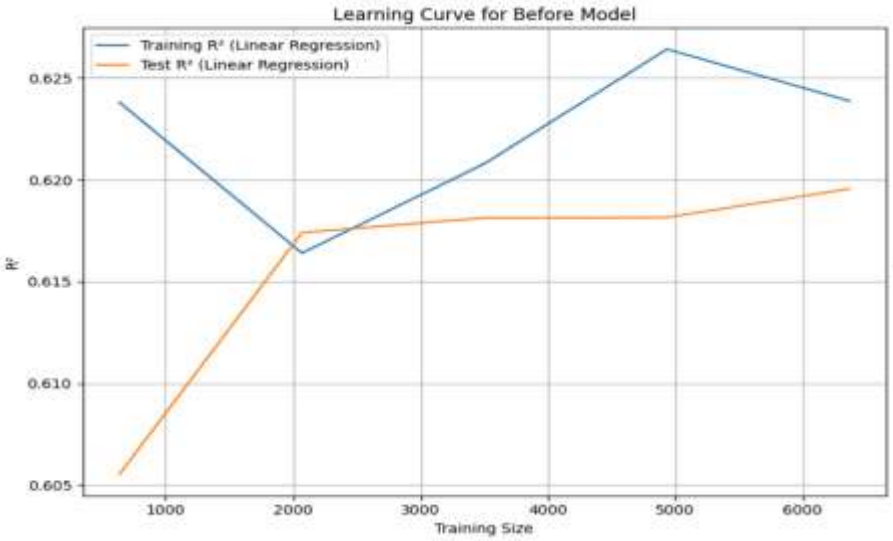
- **Hệ số tương quan cao:** cung cấp thông tin tương tự cho mô hình. Việc loại bỏ một trong các đặc trưng này có thể giúp đơn giản hóa mô hình mà không làm mất đi thông tin quan trọng
- **Hệ số tương quan thấp:** Các đặc trưng này không có mối quan hệ mạnh mẽ với nhau, cung cấp thông tin độc lập. Việc loại bỏ các đặc trưng có hệ số tương quan thấp giúp giảm độ phức tạp của mô hình

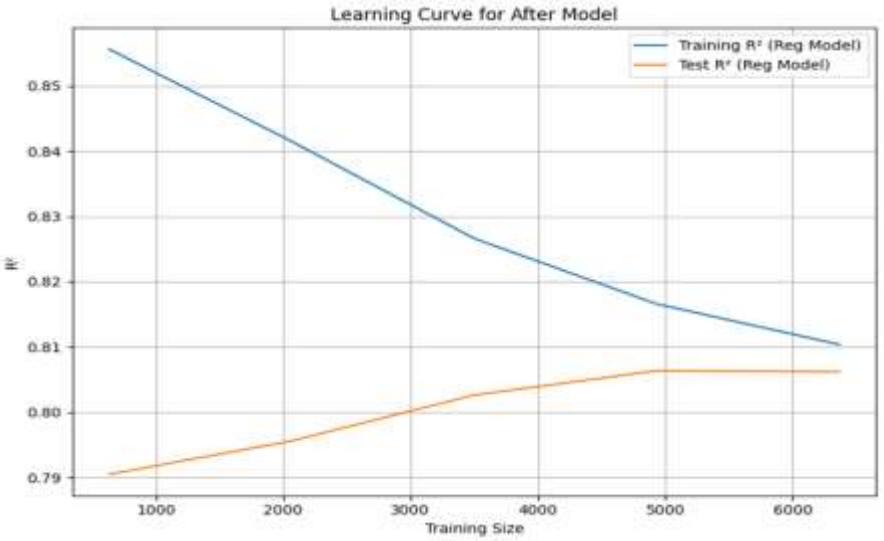
THỰC NGHIỆM CORRELATION ANALYSIS



- Các cột được giữ lại nếu bất kỳ giá trị tương quan nào của chúng nằm trong khoảng từ 0.1 đến 0.9
- Chọn 3 feature bao gồm: bathroom, bedroom, price

ĐÁNH GIÁ CORRELATION ANALYSIS





- **Trước khi train lại:** Mô hình có dấu hiệu quá khớp, với R^2 trên tập huấn luyện cao nhưng R^2 trên tập kiểm tra thấp
- **Sau khi train lại:** R^2 trên tập huấn luyện và kiểm tra tiệm cận gần nhau hơn, cho thấy khả năng tổng quát hóa tốt hơn và giảm bớt tình trạng quá khớp.

Thanks for your listening.